AD-755 118

LAGRANGEAN RELAXATION AND ITS USES IN INTEGER PROGRAMMING

Arthur M. Geoffrion

California University

Prepared for:

Office of Naval Research
National Science Foundation

December 1972

Working Paper No. 195

LAGRANGEAN RELAXATION AND ITS USES IN INTEGER PROGRAMMING

by

A.M. GEOFFRION

December 1972

WESTERN MANAGEMENT SCIENCE INSTITUTE

University of California, Los Angeles

R 64

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author) Western Management Science Institute University of California Los Angeles, California 90024	2a REPORT SECURITY CLASSIFICATION Unclassified
	2b GROUP

3 REPORT TITLE

Lagrangean Relaxation and its Uses in Integer Programming

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

Working Paper

5 AUTHOR(S) (Last name, first name, initial)

Arthur M. Geoffrion

6 REPORT DATE December 1972	7a TOTAL NO. OF PAGES ~~60~~ 64	7b NO. OF REFS 21

8a CONTRACT OR GRANT NO. N00014-69-A-0200-4042	9a ORIGINATOR'S REPORT NUMBER(S) Working Paper No. 195
b PROJECT NO. c. NR 047-041 d.	9b OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10 AVAILABILITY/LIMITATION NOTICES

Available on request through: Western Management Science Institute University of California Los Angeles, California 90024

11 SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY

13 ABSTRACT

Taking a subset of the constraints of a general mixed integer linear program up into the objective function in a Lagrangean fashion (with fixed multipliers) yields what we call a "Lagrangean relaxation" of the original program. This paper gives a reasonably comprehensive development of the use of this simple device in the context of branch- and-bound. The selective application of these ideas can yield significant improvements in performance for special classes of problems.

Security Classification

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
integer programming						
duality theory						
discrete optimization						
lagrange multipliers						
branch-and-bound						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ______."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ______."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ______."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.

Security Classification

Working Paper No. 195

Western Management Science Institute
University of California, Los Angeles

"Lagrangean Relaxation and its Uses in Integer Programming"

A. M. Geoffrion

December 1972

Presented at the IBM International Symposium on Discrete Optimization, Wildbad, Germany, October 30-November 1, 1972. This research was supported by the Office of Naval Research under Contract Number N00014-69-A-0200-4042 and by the National Science Foundation under Grants GP-26294 and GP-36090X. Reproduction in whole or in part is permitted for any purpose of the United States Government.

I-C

Abstract

Taking a subset of the constraints of a general mixed integer linear program up into the objective function in a Lagrangean fashion (with fixed multipliers) yields what we call a "Lagrangean relaxation" of the original program. This paper gives a reasonably comprehensive development of the use of this simple device in the context of branch-and-bound. The selective application of these ideas can yield significant improvements in performance for special classes of problems.

Acknowledgement

This paper has benefited from careful readings by Marshall L. Fisher and Roy E. Marsten.

Table of Contents

1. INTRODUCTION

The general integer linear programming problem can be written as

$$(\mathrm{P}) \qquad \underset{x \geq 0}{\text{Minimize}} \quad cx \quad \text{subject to} \quad \begin{array}{l} Ax \geq b \\ Bx \geq d \\ x_j \text{ integer, } j \in I , \end{array}$$

where b, c and d are vectors and A and B are matrices of conformable dimensions, and the set I consists of the subscripts of the variables required to be integer. The reason for distinguishing two groups of constraints is that the second of these, $Bx \geq d$, is supposed to have special structure. We define the <u>Lagrangean</u> <u>relaxation</u> of (P) relative to $Ax \geq b$ and a conformable nonnegative Lagrangean vector λ to be:

$$(\mathrm{PR}_\lambda) \qquad \underset{x \geq 0}{\text{Minimize}} \quad cx+\lambda(b-Ax) \quad \text{subject to} \quad \begin{array}{l} Bx \geq d \\ x_j \text{ integer, } j \in I . \end{array}$$

The fruitful application of (PR_λ) in specific cases requires judicious partitioning of the constraints into the two groups $Ax \geq b$ and $Bx \geq d$, and an appropriate choice of $\lambda \geq 0$. Three specific types of $Bx \geq d$ constraints of considerable practical importance are introduced in Sec. 1.1 and carried throughout the paper.

The purpose of this paper is to explore the usefulness of (PR_λ) as an adjunct to branch-and-bound or implicit enumeration methods for (P) . The development is intended for use at two levels. Pedagogically, it strives for a simplified and unified exposition of a number of old and new developments in integer programming. As a research effort it aims to develop what appears to be a potent general approach to the design of improved algorithms for special classes of integer programs. Although the algorithmic context of this paper is the branch-and-bound approach to integer linear programs, it should become obvious to the reader how these ideas can also be applied to other classes of algorithms and problems. For instance, a promising class of cutting-planes based on Lagrangean relaxation ideas is obtained as a part of our development.

No claim is made for the originality of the notion of Lagrangean relaxation. To the contrary, it has been used recently by Held and Karp [16] [17] in their highly successful work on the traveling-salesman problem; by Fisher [5] in his promising algorithm for scheduling in the presence of resource constraints; by Fisher and Schrage [6] in their proposed algorithm for scheduling hospital admissions; and by Shapiro [18] and Fisher and Shapiro [7] in the context of a group theoretic approach to general integer programming. No doubt other authors have also made special application of Lagrangean relaxation ideas implicitly if not explicity in their work. We should also mention the general relevance of the literature on Lagrangean methods for non-convex optimization (e.g., Brooks and Geoffrion [2] and Greenberg and Robbins [14]).

The general plan of this paper is as follows. The basic results concerning the relation between (P) and (PR_λ) are collected in Sec. 2. Duality turns out to play a surprisingly major role. In Sec. 3 a generic LP-based enumerative approach for (P) is reviewed, and the four basic uses of Lagrangean relaxation in this context are described. The basic strategic options relating to the use of (PR_λ) are also detailed. Sec. 4 describes how cutting-planes can be generated based on (PR_λ). The "filter" constraint of Balas [1], the "profit constraint" of Healy [15], and several new cuts are obtained as special cases. Sec. 5 derives the penalties of Driebeek [4] and Tomlin [19] from the viewpoint of Lagrangean relaxation, and new improved penalties are developed for important special cases. The use of penalties in strengthening the cuts of Sec. 4 and deriving new ones is also explained. In Sec. 6 the concept of surrogate constraints as developed by Glover [13] [14] and Geoffrion [8] is shown to be subsumed by the Lagrangean relaxation viewpoint. Some concluding comments are given in Sec. 7.

Three particular but important cases for the special constraints $Bx \geq d$ are carried throughout the paper. They serve to illustrate general ideas in a concrete way, and also to emphasize that Lagrangean relaxation is intended to be specialized to particular problem structures. These three cases are introduced in the following subsection. The final subsection of this Introduction summarizes the special notations and assumptions commonly used in the sequel.

1.1 Three Examples

The Lagrangean relaxation (PR_λ) must be vastly simpler to solve than (P) itself in order for it to yield any computational advantage. It should admit a closed form solution or be solvable by an efficient specialized algorithm. Thus the constraints $Bx \geq d$ must possess considerable special structure. Three typical examples of such structure will now be given. These examples will be referred to repeatedly in the sequel.

Example 1. The constraints $Bx \geq d$ specify only upper bounds on some or all of the variables. For instance, in 0-1 programming problems the integer variables possess upper bounds of unity. It is easy to see that the optimal solution of (PR_λ) can be written down by inspection of the signs of the collected coefficient vector of x, namely $(c - \lambda A)$.

Example 2. The constraints $Bx \geq d$ are as in Example 1 but also include some generalized upper bounding constraints of the form

(1) $$\sum_{j\in J_k} x_j = 1 \ , \ k = 1,2,\ldots, K \ ,$$

where $J_1,\ldots,J_K$ are disjoint subsets of I. Such constraints perform a "multiple choice" function. The optimal solution of (PR_λ) can again be written down by inspection, with a search for the smallest $(c - \lambda A)_j$ now being necessary over each subset J_k.

Example 3. The constraints $Bx \geq d$ are as in Example 1 but also include some "switching" constraints of the form

(2) $$\sum_{j\in J_k} \beta_{kj}x_j \leq \beta_{kk}x_k \ , \ k=1,\ldots,K \ ,$$

where the K subsets $\{k,J_k\}$ are disjoint, $x_1,\ldots,x_K$ are 0-1 variables, the variables in J_k are continuous-valued, and all β coefficients are strictly positive. This type of constraint typically arises in capital expansion models. In the familiar capacitated plant location problem, for example, x_k is 1 or 0 according to whether or not a plant of capacity β_{kk} is built at the kth site, x_j for $j\in J_k$ corresponds to the amounts shipped from plant site k to various destinations, and the β_{kj}'s are all unity. The Lagrangean relaxation (PR_λ) can still be solved easily, since it separates into K independent problems of the form

(3_k) $$\begin{aligned} \text{Minimize} \ & \sum_{j\in J_k} (c-\lambda A)_j x_j + (c-\lambda A)_k x_k \\ \text{subject to} \ & \sum_{j\in J_k} \beta_{kj}x_j \leq \beta_{kk}x_k \\ & 0 \leq x_j \leq u_j, j\in J_k \\ & x_k=0 \text{ or } 1 \ , \end{aligned}$$

where u_j is the upper bound on variable x_j . If $x_k=0$, it follows from the positivity of β_{kj} that $x_j=0$ must hold for all $j \in J_k$. If $x_k=1$, (3_k) becomes a trivial linear program sometimes referred to as a "continuous knapsack problem" with bounded variables; its solution is easily determined by comparison of the ratios

$$(c-\lambda A)_j/\beta_{kj} \ .$$

The best of the solutions obtained under the two cases $x_k=0$ and $x_k=1$ yields the true optimal solution of (3_k). From these K solutions one may directly assemble the optimal solution of (PR_λ).

These three examples serve to illustrate some of the commonly occuring types of special constraints $Bx \geqq d$ for which the associated Lagrangean relaxation can be optimized very efficiently. In most practical applications of integer programming there are several obvious and tractible choices for the constraints to be designated as $B\,x \geqq d$. In Held and Karp's excellent work on the traveling-salesman problem [16] [17], for example, (PR_λ) is a minimum spanning "1-tree" problem for which highly efficient algorithms are available. And in Fisher and Schrage's algorithm for hospital admissions scheduling [6], (PR_λ) separates into a relatively simple scheduling problem for each patient.

1.2 Notation and assumptions

Notation and terminology is generally standard and consistent with that of Geoffrion and Marsten [11], a survey paper containing additional background material. However, the reader should memorize the following peculiar notations: if $(\cdot)$ is an optimization problem, then $v(\cdot)$ is its optimal value, $F(\cdot)$ is its set of feasible solutions, and $(\bar{\cdot})$ refers to the same problem with all integrality conditions on the variables dropped. The vector $\bar{\lambda}$ denotes the optimal multiplier vector (dual solution) associated with the constraints $Ax \geq b$ for the ordinary linear program $(\bar{P})$.

We adopt the convention that the optimal value of an infeasible optimization problem is $+\infty$ (resp. $-\infty$) if it is a minimizing (resp. maximizing) problem. The inner product of two vectors, be they row or column, is denoted simply by their justaposition.

Two benign assumptions are made throughout this paper in the interest of decluttering the exposition, except where explicitly stated to the contrary. The first is that the non-special constraints $Ax \geq b$ are all inequality constraints. If some of these constraints were given as equalities, then the corresponding components of λ would not be required to be nonnegative. This is the only change required to accomodate equality constraints. The second assumption is that the special constraints $Bx \geq d$ include upper bounds on all variables. This obviates the need for special treatment of the case where (P) or one of its relaxations has optimal value equal to $-\infty$, and it also leads to some notational economies as explained in Sec. 3.2. This assumption is consistent with the vast majority of potential applications. It is a simple exercise to allow for its absence in the all of the results to follow..

2. THEORY OF LAGRANGEAN RELAXATION

The term <u>relaxation</u> is used in this paper in the following sense: a minimizing problem (Q) is said to be a relaxation of a minimizing problem (P) if $F(Q) \supseteq F(P)$ and the objective function of (Q) nowhere exceeds in value that of (P) on $F(P)$. Clearly (PR_λ) is a relaxation in this sense for all $\lambda \geq 0$, for the extra Lagrangean term $\lambda(b - Ax)$ in the objective function of (PR_λ) must be non-positive when $Ax \geq b$ is satisfied. Notice that the common practice of relaxation by simply throwing away some of the constraints is equivalent to Lagrangean relaxation with $\lambda = 0$. Permitting $\lambda \neq 0$ allows the relaxation to be tighter. Notice also that if an optimal solution of (PR_λ) is feasible in (P) and satisfies $\lambda(b - Ax) = 0$, then it must be optimal in (P).

The potential usefulness of any relaxation of (P), and of a Lagrangean relaxation in particular, is largely determined by how near its optimal value is to that of (P). This furnishes a criterion by which to measure the "quality" of a particular choice for λ. The ideal choice would be to take λ as an optimal solution of the concave program

$$\text{(D)} \qquad \underset{\lambda \geq 0}{\text{Maximize}} \; v(PR_\lambda) \; ,$$

which we have designated by (D) because it coincides with the formal dual of (P) with respect to the constraints $Ax \geq b$ (see, eg., [9]). This problem in turn is intimately linked to the following relaxation of (P):

$$\text{(P*)} \qquad \underset{x}{\text{Minimize}} \; cx \; \text{ subject to } \; \begin{array}{l} Ax \geq b \\ x \in Co\{x \geq 0 : Bx \geq d \text{ and} \\ x_j \text{ integer, } j \in I\} \, , \end{array}$$

where Co denotes the convex hull of a set. It may be difficult to express the convex hull in (P*) as an explicit set of linear constraints, but in principle this is always possible and so (P*) may be regarded as a linear program. An optimal multiplier vector corresponding to $Ax \geq b$ will be denoted by λ^* when (P*) has finite optimal value.

Theorem 1 collects the important relationships between (PR_λ) , $(\bar{P})$, (P^*) , (P) and (D) .

<u>Theorem 1</u>.

A. $F(\bar{P}) \supseteq F(P^*) \supseteq F(P)$, $F(PR_\lambda) \supseteq F(P)$

$v(\bar{P}) \leq v(P^*) \leq v(P)$, $v(PR_\lambda) \leq v(P)$ for all $\lambda \geq 0$.

B. If $(\bar{P})$ is feasible, then $v(\bar{P}) \leq v(PR_{\bar{\lambda}})$.

C. If (P^*) is feasible, then

$$v(D) = \underset{\lambda \geq 0}{\text{Max}} \; v(PR_\lambda) = v(PR_{\lambda^*}) = v(P^*) .$$

D. Let $(\bar{P})$ be feasible and (PR_λ) have the following <u>Integrality Property</u> for all $\lambda \geq 0$: its optimal value is not altered by dropping the integrality conditions on the variables, i.e., $v(PR_\lambda) = v(\overline{PR_\lambda})$ for all $\lambda \geq 0$. Then (P^*) is feasible and

$$v(\bar{P}) = v(PR_{\bar{\lambda}}) = v(D) = v(PR_{\lambda^*}) = v(P^*) .$$

Proof. Part A is trivial. Let $(\bar{P})$ be feasible. Then

$$\begin{aligned} v(\bar{P}) &= \max_{\lambda \geq 0} v(\overline{PR_\lambda}) && \text{by the dual theorem of linear programming}^{+/} \\ &= v(\overline{PR_{\bar{\lambda}}}) && \text{by the definition of } \bar{\lambda} \\ &\leq v(PR_{\bar{\lambda}}) && \text{because } F(\overline{PR_{\bar{\lambda}}}) \supseteq F(PR_{\bar{\lambda}}) \ . \end{aligned}$$

This proves Part B. An identical argument (the third portion is not needed) applied to (P^*) yields the conclusion of Part C if one uses the following observation in the obvious way:

$$v(PR_\lambda) = \min_x \ cx + \lambda(b - Ax) \quad \text{subj. to } x \in Co\{x \geq 0 : Bx \geq d \text{ and } x_j \text{ integer, } j \in I\} \ ,$$

which holds because the minimum value of a linear function over any compact set is not changed if the set is replaced by its convex hull. In view of Parts B and C, to prove Part D it is enough to show $v(\bar{P}) = v(P^*)$.

+/ We have taken here the "partial" dual of $(\bar{P})$ with respect to the constraints $Ax \geq b$, rather than the "full" dual customarily used in linear programming. See Geoffrion [9] (especially Sec. 6.1) for an account of this generalization of the traditional duality theory. It is easily verified that $\bar{\lambda}$ is a bona fide optimal solution of the partial dual even though it may be defined in terms of the full dual.

$$\begin{aligned} v(\bar{P}) &= \underset{\lambda \geq 0}{\text{Max}} \; v(\overline{PR_\lambda}) && \text{by duality} \\ &= \underset{\lambda \geq 0}{\text{Max}} \; v(PR_\lambda) && \text{by the Integrality Property} \\ &= \underset{\lambda \geq 0}{\text{Max}} \left[\underset{x}{\text{Min}} \; cx+\lambda(b-Ax) \;\; \text{subj. to} \;\; x \in \text{Co}\{x \geq 0: Bx \geq d \text{ and } x_j \text{ integer, } j \in I\} \right] \\ & && \text{by the observation used in the proof of Part C} \\ &= v(P^*) && \text{by duality} \; . \end{aligned}$$

Notice that the feasibility of (P*) is a consequence of the fact that its dual has finite optimal value.

QED

A few comments on the significance and interpretation of Theorem 1 are in order. Part A simply states the most obvious relations between (P) and its relaxations $(\bar{P})$, (P*) and (PR_λ) . Part B shows that $\bar{\lambda}$, an immediate byproduct of optimally solving the standard LP relaxation $(\bar{P})$, yields a Lagrangean relaxation that is at least as good as $(\bar{P})$ itself. Thus $\bar{\lambda}$ is a reasonably good feasible solution of (D). Part D shows that $\bar{\lambda}$ is in fact the optimal choice when (PR_λ) has the Integrality Property for all $\lambda \geq 0$. In this case Lagrangean relaxation can do no better than $(\bar{P})$. Part C is of interest when the Integrality Property does not hold. It shows that Lagrangean relaxation can do as well as (P*) , which in some cases may be a considerably tighter relaxation than $(\bar{P})$ --especially when the

$Ax \geq b$ constraints are not very restrictive by comparison with $Bx \geq d$. In the extreme case that $Ax \geq b$ is vacuous, evidently $v(P^*) = v(P)$.

The Integrality Property clearly holds for Example 1 and 2 above (one may assume without loss of generality that the upper bounds on the integer variables are integers), but it does not hold for Example 3. The presence or absence of the Integrality Property is evident in many applications upon inspection of (PR_λ) in light of the special structure of the constraints $Bx \geq d$. In other applications one may be able to appeal to the total unimodularity characterization of natural integer solutions of linear programming problems (e.g., Veinott and Dantzig [21]).

3. THE USE OF LAGRANGEAN RELAXATION WITH AN LP-BASED ENUMERATIVE ALGORITHM

Lagrangean relaxation can be viewed as an auxiliary device useful in connection with an LP-based algorithm for (P) . It can be used with non-LP-based algorithms as well, but its use is particularly easy and natural when the primary algorithm for (P) solves a sequence of linear programs. Since virtually all currently successful general integer linear programming algorithms are enumerative as well as LP-based (see Sec. 3.1 of [11]), we choose to describe the applications of Lagrangean relaxation in this particular context.

3.1 Generic LP-Based Enumerative Algorithm for (P)

It is necessary at this point to very briefly describe a generic LP-based enumerative algorithm for (P) . The version given here is not the most general possible, but it does comprehend the current generation of successful implementations. The description is based on portions of [11], which can be consulted for further details.

The unenumerated portion of problem (P) is represented at any given time by a list of so-called candidate problems, each of which is simply (P) with certain additional constraints appended. Each of these constraints stipulates that the value of one of the integer variables must lie in a certain closed interval (degenerate "point" intervals are allowed). When a candidate problem (CP) is examined and it is determined that it could not yield a feasible solution to (P) which is better than the best feasible solution previously found, then (CP) is said to be fathomed. The best currently known feasible solution of (P) is called the incumbent. Its cost is designated by $\underline{z}^*$. When a candidate problem is examined but not fathomed, it is separated into two simpler candidate problems (sometimes called

<u>subproblems</u>) by the addition of dichotomous interval constraints on one of the integer variables. The variable selected for this purpose is called the <u>separation variable</u>. For instance, if x_3 were the separation variable then one of the subproblems might receive the new constraint $x_3 = 0$, while the other would receive the new constraint $x_3 \geq 1$. (There are many other ways to separate a candidate problem, but this type of separation is by far the most common.)

An outline of a generic LP-based enumeratave algorithm for (P) can now be given.

<u>Step 1</u>. Initialize the list of candidate problems to consist of (P) alone and set z^* to an arbitrarily large number.

<u>Step 2</u>. Stop if the list of candidate problems is empty: if there is an incumbent then it must be optimal, otherwise (P) has no feasible solution.

<u>Step 3</u>. Choose and extract one member of the candidate problem list to become the current candidate problem (CP) .

<u>Step 4</u>. Define the current relaxed candidate problem (CP_R) to be $(\overline{CP})$ [(CP) with the conditions "x_j integer, $j \in I$" dropped].

<u>Step 5</u>. Solve (CP_R) by linear programming or some other suitable algorithm.

<u>Step 6</u>. If (CP_R) proves to be infeasible, go to Step 2.

Step 7. If the optimal value of (CP_R) proves to be no less than z^*, go to Step 2.

Step 8. If an optimal solution x^R of (CP_R) happens to be optimal in (CP) [i.e., $x^R \in F(CP)$ and $v(CP_R) = cx^R$], then record this solution as the new incumbent, put $z^* = cx^R$, and go to Step 2.

Step 9. Decide whether or not to persist in attempting to fathom (CP). If so, go to Step 10; otherwise go to Step 11.

Step 10. Replace (CP_R) by another (hopefully tighter) relaxation of (CP), and return to Step 5.

Step 11. Select a separation variable $j_o, j_o \in I$, whose optimal value $\bar{x}_{j_o}$ in $(\overline{CP})$ was not integral. Separate (CP) into two subproblems via the dichotomous interval constraints "$x_{j_o} \leq [\bar{x}_{j_o}]$" and "$x_{j_o} \geq [\bar{x}_{j_o}] + 1$", where $[\bar{x}_{j_o}]$ stands for the integer part of $\bar{x}_{j_o}$, and add the two subproblems to the candidate list.

By appropriate specializations of Steps 3, 9, 10 and 11, the above serves to describe MPSX-MIP, OPHELIE MIXED, the mixed integer option of UMPIRE, and integer programming codes developed by Beale-Small, Dakin, Davis, Davis-Kendrick-Weitzman, the author, Mitra, and others. See Sec. 3.1 of [11] for references.

3.2 Preliminary Remarks on the use of Lagrangean Relaxation

The notion of relaxation played a central role in the generic algorithm just reviewed. At Step 4 the integrality conditions on the current candidate problem are dropped, making it possible to compute an optimal solution of the relaxed candidate problem using the efficient methods of linear programming. This results in fathoming (at Step 6 or 7 or 8), or in guidance concerning how to separate the candidate problem into easier subproblems (at Step 11). The whole point is to gain access to the powerful machinery of linear programming.

Lagrangean relaxation is useful as a means of gaining access to other strong but still easily solved relaxations of the candidate problem. These relaxations can take advantage of the LP solutions generated at Step 5 in a natural way, and can be designed to exploit many common special structural features found in practical integer programming applications.

There are four basic uses of Lagrangean relaxation within the context of the generic algorithm: as a means of fathoming the current candidate problem, of guiding its separation into more tractable subproblems, of restricting the range of some of the variables prior to separation (without sacrificing the optimal solution), and of generating improved feasible solutions of (P). Each of these uses will be discussed in turn. An optional fifth use pertaining to the construction of cutting-planes (cf. Step 10) is treated separately in Sec. 4.

Applications of Lagrangean relaxation within the context of the generic algorithm require it to be applied to the candidate problems rather than to (P) itself. Our assumption that range restrictions on all variables are incorporated into the special constraints $Bx \geq d$ implies, however, that (P) and (CP) will always have the same form. This obviates the need to introduce special notation for the additional restrictions associated with (CP). We may thus use exactly the same notation for (PR_λ) and (CPR_λ), the Lagrangean relaxation of (CP), and the results of Sec. 2 apply to (CP) as well as to (P). Separation constraints more complex than the simple interval constraints we have assumed at Step 11 would require a more scrupulous notation.

The reader will notice in what follows that λ is typically taken to be $\bar{\lambda}$--a byproduct of the linear programming solution to $(\overline{CP})$. This is a natural choice, but other possible strategies for selecting λ are discussed in Sec. 3.7. Taking $\lambda=\bar{\lambda}$ also implies that Lagrangean relaxation is invoked after $(\overline{CP})$ has been solved as a linear program, presumably as a result of exercising the option of Step 10. Secs. 3.3-3.6 are written under this presumption for ease of exposition. But Sec. 3.7 discusses the frequently attractive alternative of invoking Lagrangean relaxation before $(\overline{CP})$ is solved, using a previously computed λ .

3.3 Fathoming

Suppose that (CP) could not be fathomed via the standard LP relaxation $(\overline{CP})$. Then at Step 9 one could decide to persist in the attempt to fathom (CP) by setting up the Lagrangean relaxation $(CPR_{\bar{\lambda}})$ at Step 10. Upon return to Step 5, $(CPR_{\bar{\lambda}})$ would be solved by some special method which exploits the structure of the special constraints. Step 6 would check for fathoming by infeasibility: $F(CPR_{\bar{\lambda}}) = \emptyset$. Step 7 would check for fathoming by value: $v(CPR_{\bar{\lambda}}) \geq z^*$. And Step 8 would check for fathoming by the detection of an optimal solution of (CP): the optimal solution of $(CPR_{\bar{\lambda}})$, say x^R, would be optimal in (CP) if it is feasible in (CP) and $\bar{\lambda}(b-Ax^R) = 0$.

Theorem 1.D sheds some light on the probable fathoming effectiveness of $(CPR_{\bar{\lambda}})$ when used as described as a supplement to the standard relaxation $(\overline{CP})$. It reveals that the tests for fathoming by infeasibility (Step 6) or by value (Step 7) cannot be effective when applied to $(CPR_{\bar{\lambda}})$ if (CPR_λ) has the

Integrality Property for all $\lambda \geq 0$ (as is the case, it will be recalled, for Examples 1 and 2). This is because $v(CPR_{\overline{\lambda}})$ must equal $v(\overline{CP})$. Step 8, on the other hand, can succeed for $(CPR_{\overline{\lambda}})$ even though it fails for $(\overline{CP})$.

3.4 Guiding separation

Separation of (CP) is necessary when the attempts to fathom it have failed. The Lagrangean relaxation $(CPR_{\overline{\lambda}})$ can be useful at Step 11 in selecting a separation variable j_o and in predicting which of the two resulting subproblems is more likely to contain the better feasible solution of (P). This involves computing the <u>conditional bounds</u>

$$(4)\qquad \begin{aligned} V_D(j) &\overset{\Delta}{=} v(CPR_{\overline{\lambda}}\,|\,x_j \le [\overline{x}_j]) \\ V_U(j) &\overset{\Delta}{=} v(CPR_{\overline{\lambda}}\,|\,x_j \ge [\overline{x}_j] + 1) \end{aligned}$$

for all indices j in I_f, defined to be the subset of I corresponding to variables that are fractional in the LP solution $\overline{x}$ of $(\overline{CP})$. The "|" notation indicates that the constraint following it (a simple range restriction on a single variable) is appended to $(CPR_{\overline{\lambda}})$. Clearly $(CPR_{\overline{\lambda}}|x_j \le [\overline{x}_j])$ is a legitimate Lagrangean relaxation of $(CP|x_j \le [\overline{x}_j])$, and similarly for $(CPR_{\overline{\lambda}}|x_j \ge [\overline{x}_j] + 1)$. Thus the numbers $V_D(j)$, and $V_U(j)$ respectively represent conditional lower bounds on the optimal values of the new "downward" and "upward" subproblems which would result if (CP) were separated on the jth variable.

As an example of the use of conditional bounds, the separation variable could be chosen so as to maximize the larger of $V_D(j)$ and $V_U(j)$ over all eligible j. Several successful integer programming codes have employed an analagous criterion applied to LP-based "penalties" -- a concept which, as we shall see in Section 5, is intimately related to (4). Once a separation variable j_o is selected, the smaller of $V_D(j_o)$ and $V_U(j_o)$ indicates which of the two resulting subproblems appears likely to contain the best feasible solution of (P). This information is useful for deciding which of the two subproblems should be selected from the candidate list first at a future execution of Step 3.

The usefulness of $(\mathrm{CPR}_{\overline{\lambda}})$ as a guide to separation depends primarily on how well $V_D(j)$ and $V_U(j)$ correlate with the true optimal values of the corresponding candidate subproblems. For Examples 1 and 2, unfortunately, it turns out that

$$(5) \qquad v(\overline{\mathrm{CP}}) = V_D(j) = V_U(j) \quad \text{for all } j \in I_f$$

if (4) is interpreted literally (more on this in Section 5). This fact and the general situation to be expected when the Integrality Property holds are the subject of Theorem 2.

<u>Theorem 2</u>

Let (CPR_{λ}) have the Integrality Property for all $\lambda \geq 0$, and let $(\overline{\mathrm{CP}})$ have a finite optimal value. Then for all $j \in I_f$

$$(6) \qquad v(\overline{\mathrm{CP}}) = v(\mathrm{CPR}_{\overline{\lambda}}) \leq \begin{cases} V_D(j) \\ V_U(j) \end{cases} \quad \text{with equality holding in at least one of the two cases.}$$

In the special situation of Examples 1 and 2, equality holds in (6) for <u>both</u> $V_D(j)$ and $V_U(j)$.

<u>Proof</u>

The equality in (6) is a direct consequence of Theorem 1.D. The second relation follows from the fact that the two conditions "$x_j \leq [\overline{x}_j]$" and "$x_j \geq [\overline{x}_j] + 1$" are mutually exclusive and exhaustive (since x_j is required to be integer-valued).

The strengthened conclusion for Examples 1 and 2 follows from the fact that $\overline{x}$ and $\overline{\lambda}$ must satisfy the optimality conditions for $(\overline{\mathrm{CP}})$ written so as to introduce a multiplier vector only for the constraints $Ax \geq b$:

(i) $\bar{x}$ minimizes $cx + \bar{\lambda}(b-Ax)$ subject to $Bx \geq d$, $x \geq 0$

(ii) $\bar{\lambda} \geq 0$

(iii) $\bar{\lambda}(b-A\bar{x}) = 0$

(iv) $A\bar{x} \geq b$.

We need only condition (i). Consider now the objective coefficient $(c-\bar{\lambda}A)_{\hat{j}}$ for $\hat{j} \in I_f$. For Example 1 this coefficient must be 0, for otherwise the term $(c-\bar{\lambda}A)_{\hat{j}}x_{\hat{j}}$ could be reduced by letting $x_{\hat{j}}$ depart from $\bar{x}_{\hat{j}}$ in the opposite direction from the sign of $(c-\bar{\lambda}A)_{\hat{j}}$. The same is true for Example 2 if $\hat{j}$ is not a member of any J_k set; and if $\hat{j}$ is a member of some J_k, then there must be a tie in the smallest objective function coefficient amongst the free variables in the same J_k subset as $x_{\hat{j}}$, for otherwise the term $\Sigma_j (c-\bar{\lambda}A)_j x_j$ (sum over the free variables just specified) could be reduced by increasing (decreasing) $x_{\hat{j}}$ if $(c-\bar{\lambda}A)_{\hat{j}}$ is (is not) the uniquely smallest coefficient. The conclusions regarding $(c-\bar{\lambda}A)_{\hat{j}}$ in Examples 1 and 2 imply that $x_{\hat{j}}$ can be forced to either $[\bar{x}_j]$ or $[\bar{x}_j] + 1$ without disturbing the optimal value of $(CPR_{\bar{\lambda}})$. The desired conclusion (5) now follows. Q.E.D.

Thus Lagrangean relaxation with $\lambda = \bar{\lambda}$ appears to be useless for guiding separation in Examples 1 and 2, since (5) gives no discrimination whatever among possible separation variables or between the resulting subproblems. The general case in which the Integrality Property holds is perhaps a little more promising, but at least one of the bounds $V_D(j)$ and $V_U(j)$ is stuck at $v(\overline{CP})$ for each $j \in I$. [We remark in anticipation of Section 5 that the situation can be significantly improved if (4) is computed using an alternative representation of x_j in terms of the non-basic variables in the optimal solution of $(\overline{CP})$.]

Lagrangean relaxation is much more promising at Step 11 when the Integrality Property does not hold. Then both $V_D(j)$ and $V_U(j)$ may exceed $v(\overline{CP})$ and may thereby provide useful discrimination among possible separation variables and the resulting candidate subproblems. Example 3 illustrates nicely how this can be, and is worth examining in some detail. Suppose for simplicity that all variables are in $\cup_{k=1}^{K} \{k, J_k\}$. Consider the initial candidate problem (CP) = (P) [an analysis similar to the one below applies to any subsequent candidate problem]. The optimal value of (3_k) with $\lambda = \bar{\lambda}$ and with x_k fixed is denoted by $v_k(x_k)$. It is known from the elementary theory of parametric linear programming that $v_k(x_k)$ is piecewise-linear and convex as a function of x_k. A graph of this function might look as in Figure 1. In this picture $v_k(1)$ happens to be greater than zero, although it could also be

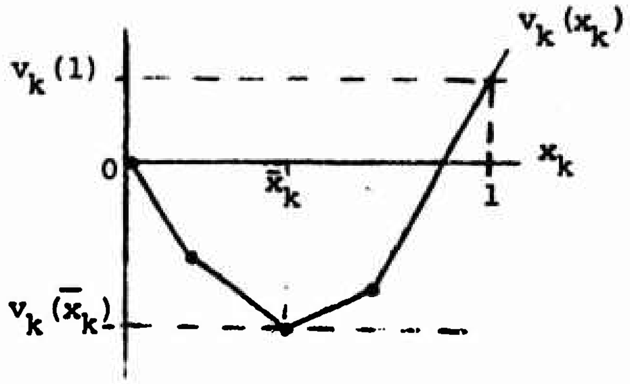

Figure 1

Hypothetical Graph of the Function $v_k(\cdot)$

less than or equal to zero. Note that $v_k(0)$ necessarily equals zero, as is evident upon inspection of (3_k). The minimum of the function is shown as occurring at $\bar{x}_k$. This is true because $(\bar{x}, \bar{\lambda})$ must satisfy the optimality conditions for $(\bar{P})$ stated as in the proof of Theorem 2. The separability of these conditions implies that $\bar{x}_k$ and $\bar{x}_j$, $j \in J_k$, are optimal in (3_k) with $\lambda = \bar{\lambda}$ and with the integrality requirement on x_k dropped -- and hence that

$$v_k(\bar{x}_k) = \underset{0 \leq x_k \leq 1}{\text{Minimum}} \; v_k(x_k)$$

It also follows from the optimality conditions that

$$(7) \qquad v(\bar{P}) = \bar{\lambda}b + \Sigma_{k=1}^{K} v_k(\bar{x}_k).$$

From the introductory discussion of Example 3 we have

$$(8) \qquad v(PR_{\bar{\lambda}}) = \bar{\lambda}b + \Sigma_{k=1}^{K} v(3_k) .$$

Clearly

$$v(3_k) = \text{Min} \left\{0, \; v_k(1)\right\} \geq v_k(\bar{x}_k).$$

We can now see plainly from (7) and (8) that $v(\bar{P}) \leq v(PR_{\bar{\lambda}})$ and that any difference is exactly equal to

$$(9) \qquad \Sigma_{k=1}^{K} \left[\text{Min} \left\{0, \; v_k(1)\right\} - v_k(\bar{x}_k)\right] .$$

This expression has a simple interpretation in terms of Figure 1. For obvious reasons there can be no contribution to this difference for the terms such that $\bar{x}_k$ happens to be 0 or 1, but for k such that $\bar{x}_k$ is fractional it is very likely that there will be a contribution.

Thus, we see why $v(P) < v(PR_{\overline{\lambda}})$ is likely whenever there are fractional integer variables in the LP solution of $(\overline{P})$ for Example 3. In this event both $v_D(k)$ and $v_U(k)$ obviously must exceed $v(\overline{P})$. The magnitude of these conditional bounds are easily interpreted in terms of graphs like Figure 1. For the particular case of Figure 1,

$$v_D(k) = v(PR_{\overline{\lambda}})$$

$$v_U(k) = v(PR_{\overline{\lambda}}) + v_k(1) .$$

3.5 Range Restriction

Sometimes during the attempted fathoming or separation of a candidate problem it is discovered that some of the range restrictions on the variables can be reduced without losing an optimal solution of (P). The range restrictions may then be tightened and the algorithm continues as before. With zero-one variables, this is sometimes known as making a "forced choice" of a variable.

Lagrangean relaxation can assist in making such discoveries when the conditional bounds of (4) are computed. If $V_D(j) \geq z^*$ happens to hold, then the lower range restriction on x_j obviously can be tightened to $x_j \geq [\overline{x}_j] + 1$. Similarly, $V_U(j) \geq z^*$ implies that the upper range restriction on x_j can be tightened to $x_j \leq [\overline{x}_j]$. It is even possible that both $V_D(j) \geq z^*$ and $V_U(j) \geq z^*$ hold, in which case it is clear the current candidate problem is <u>fathomed</u>.

Similar reasoning can be applied to $j \in I$ even if $\overline{x}_j$ is integer:

$$v(CPR_{\overline{\lambda}} | x_j \leq \overline{x}_j - 1) \geq z^*$$

implies that $x_j \geq \overline{x}_j$ can be imposed, and

$$v(CPR_{\overline{\lambda}} | x_j \geq \overline{x}_j + 1) \geq z^*$$

implies that $x_j \leq \overline{x}_j$ can be imposed. Note that either of these situations can occur even though $v(CPR_{\overline{\lambda}}) < z^*$.

All of the conclusions reached in the previous section regarding the magnitudes of the conditional bounds $V_D(j)$ and $V_U(j)$ are of direct interest with regard to range restriction.

3.6 Generating Improved Feasible Solutions

Recall from Section 3.3 that $(CPR_{\overline{\lambda}})$ leads to the fathoming of (CP) at Step 8 if the optimal solution x^R of $(CPR_{\overline{\lambda}})$ is feasible in (CP) and satisfies $\overline{\lambda}(b-Ax^R) = 0$. If x^R is feasible in (CP) but $\overline{\lambda}(b-Ax^R) \neq 0$, then it is worth checking whether $cx^R < z^*$. If so, x^R should be recorded as a new incumbent and z^* should be updated even though (CP) is not fathomed.

Step 8 can be broadened still further if x^R is not feasible in (CP). Adjusting x^R in some ad hoc way so as to recover feasibility may well lead to an improved feasible solution of (P). This is exactly the same tactic as is commonly used with the ordinary LP relaxation $(\overline{CP})$, where the LP solution is rounded to satisfy integrality in the hope that it will become feasible in (P) and have a value better than that of the incumbent.

The effective design of feasible solution generators along these lines usually must be tailored to the specific problem structure at hand.

3.7 Strategic Options for the Use of Lagrangean Relaxation

The exposition of the preceding subsections presumes for the most part that Lagrangean relaxation is invoked only after

$(\overline{CP})$ is solved, and that λ is always taken to be the $\bar{\lambda}$ corresponding to $(\overline{CP})$. This might be called the <u>post-LP strategy</u> with the natural choice for λ. It is important to recognize that there is also what might be called a <u>pre-LP strategy</u>, and that the candidate problem counterparts of (D) and (P^*) may yield better choices of λ than $(\overline{CP})$ when the Integrality Property does not hold for (CPR_λ).

The idea of the pre-LP strategy is to invoke (CPR_λ) just <u>before</u> $(\overline{CP})$ is to be solved. The hope is that (CP) can thereby be fathomed without having to resort to the more expensive linear program $(\overline{CP})$. The natural choice for λ is to use a multiplier vector saved from a previous execution of Step 5 at which the standard LP relaxation of a candidate problem was addressed. The λ used should preferably be the one corresponding to the prior candidate problem most closely related to the current one; or, if this proves cumbersome, simply the last one generated. Notice that it is particularly worth saving for this purpose the multiplier vector for a candidate problem which is fathomed by value via its LP relaxation [$v(CPR_{\bar{\lambda}}) \geq z^*$ necessarily holds for this (CP) and its associated $\bar{\lambda}$]. The computational experience cited in Section 3.1.5 of [11] (see also Geoffrion [8]) shows that the pre-LP strategy can be quite effective even in the context of Example 1.

The other point we wish to elaborate upon is that, for both the pre-LP and post-LP strategies, it is possible to use a λ other than an optimal multiplier vector associated with the usual LP relaxation of a candidate problem. One may wish to do this, for instance, when $(\overline{CP})$ appears to be excessively expensive to solve for a truly optimal solution but techniques are available for generating a reasonably good suboptimal multiplier vector. Fisher [5] has used this variant quite successfully in the context of scheduling in the presence of resource constraints. Here $(\overline{CP})$ has such a large number of columns that it is impractical to even write down explicitly much less solve optimally. Nevertheless, he was able to work out a means of implementing what is essentially the dual Simplex Method for finding good dual solutions at reasonable computational cost.

A second reason for using λ other than $\overline{\lambda}$ is evident from Theorem 1, which shows that better choices than $\overline{\lambda}$ may exist when (CPR_{λ}) does not have the Integrality Property for all $\lambda \geqq 0$. One then looks to the candidate problem counterparts of (D) or (P^{*}) for superior λ's. If available, $\overline{\lambda}$ furnishes a good starting point to improve upon. One possibility is to apply an ascent technique to the counterpart of (D), which is a concave program, so as to obtain at least a near-optimal solution. This has been done by Held and Karp [17] and by Fisher and Shapiro [7]. Another possibility is to apply a Dantzig-Wolfe decomposition technique to the counterpart of (P^{*}), with the convex hull therein represented in terms of its extreme points (cf. Section 3 of Held and Karp [16] and Brooks and Geoffrion [2]). The column-generation problems would be precisely of the form (PR_{λ}). The calculations would very likely be terminated before complete optimization, due to the characteristically slow asymptotic convergence of this technique. Other specialized techniques, both exact/heuristic, can be devised for (D) or (P^{*}) in particular applications. We feel that this is likely to be a fruitful area of investigation when engineering a computer code for (P).

4. CUTS BASED ON LAGRANGEAN RELAXATION

For present purposes a cut is any linear inequality defined relative to a candidate problem which must be satisfied by all of its feasible solutions. Generally we seek cuts which, when appended to the standard LP relaxation $(\overline{CP})$ of a candidate problem, cause the optimal value of $(\overline{CP})$ to increase as much as possible. It is therefore desirable for a cut to have the property that it is not satisfied by the available optimal solution to $(\overline{CP})$. Cuts are one of the most important ways of carrying out Step 10 of the generic enumerative approach described earlier, that is, of tightening an LP relaxation of an integer program. We shall see that the notion of Lagrangean relaxation leads quite directly to a remarkably flexible family of cuts.

To be specific we shall consider the initial candidate problem only, namely (P) itself. It is a simple matter to see how the ideas developed below can be applied to any candidate problem.

4.1 Introduction to Lagrangean Cuts

The fundamental idea for generating cuts based on (PR_λ) for any specific $\lambda \geq 0$ is this. Select a subset of the variables of (P), say $\{x_1,\ldots,x_{n_1}\}$, for which it is possible to determine an explicit linear function $\ell(x_1,\ldots,x_{n_1})$ satisfying

(10) $$\ell(x_1,\ldots,x_{n_1}) = v(PR_\lambda | x_1,\ldots,x_{n_1})$$

for all vectors $(x_1\ldots,x_{n_1})$ which are part of a ~~some~~ feasible solution of (PR_λ). Here $v(PR_\lambda | x_1,\ldots,x_{n_1})$ is intended as a functional notation referring to the optimal value of

(PR_λ) with the additional restriction that the specified variables take on the specified values. (If a particular vector $(x_1,\ldots,x_{n_1})$ cannot be completed so as to yield a feasible solution of (PR_λ), then by convention $v(PR_\lambda | x_1,\ldots,x_{n_1})$ is defined to be $+\infty$ at this point.) The cut deriving from (10) is

$$(11) \qquad \ell(x_1,\ldots,x_{n_1}) \le cx \quad .$$

This is a legitimate cut because

$$\ell(x_1,\ldots,x_{n_1}) = v(PR_\lambda | x_1,\ldots,x_{n_1}) \le v(P | x_1,\ldots,x_{n_1}) \le cx$$

holds for all x feasible in (P).

At least two choices for $\{x_1,\ldots,x_{n_1}\}$ are always permissible regardless of the particular structure of (PR_λ) : the empty set and the full set. For the empty set the function ℓ is just the constant $v(PR_\lambda)$. Thus (11) becomes simply

$$(12) \qquad v(PR_\lambda) \le cx \quad .$$

Note that (12) cannot be violated by $\bar{x}$, an optimal solution of $(\bar{P})$, when (PR_λ) has the Integrality Property for all $\lambda \geq 0$. In that case $v(PR_\lambda) \le v(\bar{P}) = c\bar{x}$ must hold by Theorem 1.D, and so appending (12) to $(\bar{P})$ cannot increase its optimal value. Theorem 1.D also reveals that $\bar{\lambda}$ is the best choice of λ in this case, since (12) must then hold with exact equality at $\bar{x}$. In the absence of the Integrality Property, (12) will be violated at $\bar{x}$ if $v(\bar{P})<v(P^*)$ and λ is sufficiently near optimal in (D).

Now consider the choice where $\{x_1,\ldots,x_{n_1}\}$ consists of all of the variables. Then

$$v(PR_\lambda|x) = cx + \lambda(b - Ax)$$

for all x feasible in (PR_λ) and (11) becomes

$$(13) \qquad cx + \lambda(b-Ax) \leq cx \quad \text{or} \quad \lambda(b-Ax) \leq 0 \ .$$

This cut obviously cannot be violated by $\bar{x}$ or by any other feasible solution of $(\bar{P})$, since then $b - Ax \leq 0$. Thus (13) is ineffective as a means of tightening the relaxation $(\bar{P})$. We remark in passing that (13) with λ equal to $\bar{\lambda}$ is precisely the "filter" constraint proposed by Balas [1] in the context of Example 1 (as noted in the proof of Theorem 2, $\bar{\lambda}(b-A\bar{x}) = 0$).

The particular variables which play the role of $\{x_1,\ldots,x_{n_1}\}$ will be called the cut variables, and of course they need not be the first n_1 consecutive components of x . Other choices besides the empty set and the full set are usually permissible in that they meet the requirement of (10) that $v(PR_\lambda|x_1,\ldots,x_{n_1})$ must be linear over the set of vectors $(x_1,\ldots,x_{n_1})$ which are part of a feasible solution of (PR_λ) . Among the permissible choices, one seeks to maximize the depth of the resulting cut in some sense. Since we have in mind using (11) at Step 10 of the generic enumerative algorithm, one appealing criterion for measuring "depth" is the amount of violation of (11) evaluated at $\bar{x}$:

$$\ell(\bar{x}_1,\ldots,\bar{x}_{n_1}) - c\bar{x} \quad . \tag{14}$$

If there are several permissible choices of cut variables which maximize the cut depth then it seems reasonable to turn to a secondary criterion of having as many cut variables as possible since this tends to strengthen the cut when evaluated at any feasible solution of (PR_λ) . This is evident from the obvious inequality

$$v(PR_\lambda|x_1,\ldots,x_{n_1}) \le v(PR_\lambda|x_1,\ldots,x_{n_1},x_{n_1+1}) \quad .$$

The other principal question arising in applications is the choice of λ . The cut depth criterion (14) serves as well to discriminate among alternative choices of λ as it does among alternative choices of cut variables. The most conspicuous choice is certainly $\bar{\lambda}$, though one must allow that another choice may prove superior -- especially when the Integrality Property does not hold.

Let us consider now our three examples with the choice $\lambda = \bar{\lambda}$.

4.2 Cuts for Example 1

For example 1 it turns out that <u>any</u> subset of cut variables is permissible. This is due to the fact that $(PR_{\bar{\lambda}})$ separates into as many independent univariate subproblems as there are components in x . Evidently

$$v(PR_{\bar{\lambda}}|x_1,\ldots,x_{n_1}) = v(PR_{\bar{\lambda}}|x_1=\cdots=x_{n_1}=0)+\sum_{j=1}^{n_1}(c-\bar{\lambda}A)_j x_j \tag{15}$$

for every $(x_1,\ldots,x_{n_1})$ which is part of a feasible solution to $(PR_{\bar{\lambda}})$. Thus (11) becomes

$$(16) \qquad v(PR_{\bar{\lambda}} | x_1=\ldots=x_{n_1}=0) + \sum_{j=1}^{n_1} (c-\bar{\lambda}A)_j x_j \leq cx \quad .$$

The depth of this cut at $\bar{x}$ is zero because

$$v(PR_{\bar{\lambda}} | x_1=\ldots=x_{n_1}=0) + \sum_{j=1}^{n_1} (c-\bar{\lambda}A)_j \bar{x}_j = v(PR_{\bar{\lambda}}) = c\bar{x}$$

by Theorem 1.D. Since all cuts have the same depth by criterion (14), the proper choice is to take $\{x_1,\ldots,x_{n_1}\}$ to be the full set. Then (16) coincides with (13) with $\lambda=\bar{\lambda}$.

4.3 Cuts for Example 2

For Example 2 it also turns out that any choice of cut variables is permissible, and that all of the resulting cuts have exactly zero depth at $\bar{x}$. We omit the straightforward but somewhat tedious verification of this result. Thus (13) with $\lambda=\bar{\lambda}$ is again the best cut.

4.4 Cuts for Example 3

The situation for Example 3 is a little more complex. Recall that $(PR_{\bar{\lambda}})$ separates into K independent subproblems of the form (3_k) and into as many additional univariate subproblems as there are variables which do not appear in any of the subproblems of type (3_k). Denote the subscripts of these last variables by T. Then

$$v(PR_{\bar{\lambda}}) = \bar{\lambda}b + \sum_{k=1}^{K} v(3_k) + \sum_{j\in T} \left[\text{Min } (c-\bar{\lambda}A)_j x_j \text{ subj. to } 0\leq x_j \leq u_j \text{ and } x_j \text{ integer if } j\in I \right].$$

Consider selecting for the cut variables any subset of the 0-1 switching variables, say $x_1,\ldots,x_{K_1}$ where $1 \leq K_1 \leq K$, plus any subset S of T.

Evidently

(17) $v(PR_{\bar{\lambda}} | x_1,\ldots,x_{K_1}; x_j \text{ for } j \in S) =$

$$v(PR_{\bar{\lambda}} | x_k=0 \text{ for } 1 \leq k \leq K_1 \text{ and } x_j=0 \text{ for } j \in S)$$

$$+ \sum_{k=1}^{K_1} v(\mathfrak{Z}_k | x_k) + \sum_{j \in S} (c-\bar{\lambda}A)_j x_j$$

holds whenever x_k is 0 or 1 for $1 \leq k \leq K_1$ and $0 \leq x_j \leq u_j$ for $j \in S$ and also x_j is integer for $j \in I \cap S$. The binary nature of x_k makes it possible to write down a linear function of x_k which coincides with $v(\mathfrak{Z}_k | x_k)$ at $x_k = 0$ and 1:

(18) $$v(\mathfrak{Z}_k | x_k) = v(\mathfrak{Z}_k | 0)(1-x_k) + v(\mathfrak{Z}_k | 1)x_k \quad .$$

Together (17) and (18) yield (uniquely) the linear function required by (10), and (11) becomes

(19) $$v(PR_{\bar{\lambda}} | x_k=0, 1 \leq k \leq K_1, \text{ and } x_j = 0 \text{ for } j \in S)$$
$$+ \sum_{k=1}^{K_1} v(\mathfrak{Z}_k | 1) x_k + \sum_{j \in S} (c-\bar{\lambda}A)_j x_j \leq cx \ ,$$

where we have used the fact that $v(\mathfrak{Z}_k | 0) = 0$. It is clear from the discussion following Fig. 1 in Sec. 3.4 that this cut is quite likely to be violated at $\bar{x}$:

(20) left-hand side of (19) evaluated at $\bar{x}$

$$= v(PR_{\bar{\lambda}} | x_k=0, 1\leq k\leq K_1) + \sum_{k=1}^{K_1} v(3_k|1)\bar{x}_k$$

$$\geq v(PR_{\bar{\lambda}} | x_k=0, 1\leq k\leq K_1) + \sum_{k=1}^{K_1} v(3_k)$$

$$= v(PR_{\bar{\lambda}}) \geq v(\bar{P}) = c\bar{x} \quad .$$

The amount of violation (i.e., the depth of cut) is thus at least equal to $v(PR_{\bar{\lambda}}) - v(\bar{P})$, which is given by (9). Figure 1 shows clearly that additional violation will accrue for each $\bar{x}_k (1\leq k\leq K_1)$ which is fractional, in the amount by which $v(3_k|1)\bar{x}_k$ exceeds $v(3_k)$. It follows that it is best to take $K_1 = K$, since this includes as many fractional $\bar{x}_k$'s as possible.

This analysis of Example 3 may be summarized by saying that a Lagrangean cut can be obtained for any subset of cut variables which does not include any of the variables in $\cup_{k=1}^{K} J_k$, and that the deepest cuts are obtained when the cut variables include all $k(1\leq k\leq K)$ such that $\bar{x}_k$ is fractional. Attempts to include some of the variables in $\cup_{k=1}^{K} J_k$ as cut variables will not always be successful since they tend to destroy the essential linearity property associated with (10). In general, then, the appropriate choice for the cut variables is $\{1,\ldots,K\}\cup T$. The corresponding cut is

(21) $$\sum_{k=1}^{K} v(3_k|1)x_k + \sum_{j\in T} (c-\bar{\lambda}A)_j x_j \leq cx \quad .$$

4.5 Further remarks

The discussion of Examples 1-3 made extensive use of the fact that (PR_λ) could be separated into a number of independent subproblems. Such separability is sufficiently common in potential applications of Lagrangean relaxation to warrant the following observation. Suppose that the component indices of x can be partitioned into K mutually exclusive and exhaustive subsets $J_1,\ldots,J_K$ ($K=1$ is allowed) such that (PR_λ) separates into K corresponding independent subproblems (PR_λ^k), $k=1,\ldots,K$ [this requires, of course, that none of the constraints $Bx \geqq d$ link any of the subsets of variables]. Then

$$v(PR_\lambda) = \lambda b + \sum_{k=1}^{K} v(PR_\lambda^k).$$

Let j_k be any particular element from J_k and it follows that

$$v(PR_\lambda | x_{j_1},\ldots,x_{j_K}) = \lambda b + \sum_{k=1}^{K} v(PR_\lambda^k | x_{j_k}) \ .$$

If $x_{j_1},\ldots,x_{j_K}$ are selected so that they are all 0-1 variables then they constitute a permissible set of cut variables. This is because [cf. (18)] the linear function

$$v(PR_\lambda^k | x_{j_k}=0)(1-x_{j_k}) + v(PR_\lambda^k | x_{j_k}=1)x_{j_k}$$

coincides with $v(PR_\lambda^k | x_{j_k})$ for $x_{j_k} = 0$ and 1, and consequently (11) becomes

$$(22) \quad \lambda b + \sum_{k=1}^{K} \left[v(PR_\lambda^k | x_{j_k}=0)(1-x_{j_k}) + v(PR_\lambda^k | x_{j_k}=1)x_{j_k} \right] \leq cx \ .$$

This cut has the property that it is always at least as deep at $\bar{x}$ as (12), the cut based on the empty set of cut variables; that is, the left-hand side of (22) evaluated at $\bar{x}$ is at least as large as $v(PR_\lambda)$. This fact rests on the simple observation

$$v(PR_\lambda^k | x_{j_k}=0)(1-\bar{x}_{j_k}) + v(PR_\lambda^k | x_{j_k}=1)\bar{x}_{j_k} \geq$$

$$\text{Min } \{v(PR_\lambda^k | x_{j_k}=0), v(PR_\lambda^k | x_{j_k}=1)\} = v(PR_\lambda^k).$$

It is perhaps unnecessary to remark that x_{j_k} need not be a 0-1 variable when J_k is a singleton in order for it to be a permissible choice for a cut variable, for then $v(PR_\lambda^k | x_{j_k}) = (c-\lambda A)_{j_k} x_{j_k}$ itself is linear.

Two further remarks are in order. First, choices of cut variables $\{x_1,\ldots,x_{n_1}\}$ which do not satisfy the linearity pro- perty of (10) can sometimes be accomodated by allowing $\leq$ instead of equality in (10). It is immediate that (11) remains a legitimate cut. Second, (11) can sometimes be improved by replacing $v(PR_\lambda | x_1,\ldots,x_{n_1})$ for some vectors $(x_1,\ldots,x_{n_1})$ by a better lower bound on $v(P|x_1,\ldots,x_{n_1})$. These two remarks are exploited in Sec. 5.5 below.

Sec. 5.5 also shows how one may overcome the apparently unsatisfactory situation for Examples 1 and 2 in which there seems to be no way to construct a cut of positive depth at $\bar{x}$.

5. RELATIONS TO THE PENALTY CONCEPT

The so-called "penalty" concept in integer programming was propelled to prominence by Driebeek [4], although the essential notion was used earlier by Healy [15] and Dakin [3]. The original idea was to underestimate the amount by which the optimal value of the LP relaxation of the current candidate problem would increase if separation were carried out using a particular separation variable. The estimates of change, referred to as penalties, can be used to help guide separation and may also permit fathoming or range restriction. An important subsequent refinement of this original idea was the recognition that computing penalties based on LP relaxations does not come to grips with the fact that it is the candidate problems and subproblems themselves, and not their LP relaxations, which are central to the underlying enumerative process. Tomlin [19] [20] showed how to modify the penalty formulae so as to take at least partial account of the integrality conditions. The resulting penalties are underestimates of the difference between $v(\overline{CP})$ and the optimal value of a candidate subproblem derived from (CP). Perhaps needless to say, in general it is not possible to take complete account of all integrality conditions in computing these strengthened penalties, for that would require just about as much work as solving (CP) itself. See [11] for a discussion of current practice in the computation and use of penalties.

Lagrangean relaxation furnishes a convenient setting for deriving the simple and strengthened penalties alluded to above. This is done in Sec. 5.1. More importantly, it leads naturally to extensions, specializations, and additional uses for penalties which do not follow as easily from the more traditional viewpoint. This is illustrated in Secs. 5.2-5.5, the last of which is devoted to the promising topic of using penalties to strengthen the Lagrangean cuts of Sec. 4.

5.1 Basic Results: $Bx \geq d$ Vacuous

The first task is to show how the formulae for simple and strengthened penalties are related to Lagrangean relaxation. This requires specializing $Bx \geq d$ to be vacuous (in contrast to our usual convention, in this subsection $Bx \geq d$ will not even include upper bounds on the variables). It follows, as in Theorem 2, that the objective function coefficient of $(CPR_{\bar{\lambda}})$ vanishes for all j such that $\bar{x}_j$ is fractional, and hence that for all $j \in I_f$ we have

$$v_D(j) \triangleq v(CPR_{\bar{\lambda}} | x_j \leq [\bar{x}_j]) = v(\overline{CP})$$

$$v_U(j) \triangleq v(CPR_{\bar{\lambda}} | x_j \geq [\bar{x}_j] + 1) = v(\overline{CP}) .$$

Thus the Lagrangean relaxation $(CPR_{\bar{\lambda}})$ appears to yield zero "down" and "up" penalties for separation on x_j .

A simple remedy is to employ an alternative representation for x_j in terms of variables whose objective function coefficients in $(CPR_{\bar{\lambda}})$ do not vanish. Such a representation is available from the final tableau of the linear program $(\overline{CP})$ since x_j must be basic therein. To facilitate comparison with prior work on penalties, assume for the remainder of this subsection that the constraints $Ax \geq b$ are actually equalities rather than inequalities (add slack variables if necessary). Then x_j has a representation in terms of the variables which are non-basic $(i \in N)$ in the final LP tableau of the form

$$x_j = \bar{x}_j - \sum_{i \in N} \bar{a}_{ji} x_i .$$

The use of this representation in the definition of $v_D(j)$ and $v_U(j)$ leads to the following strengthened conditional bounds:

for $j \in I_f$,

(23)
$$V_D^*(j) \triangleq v(CPR_{\overline{\lambda}} | \overline{x}_j - \sum_{i \in N} \overline{a}_{ji} x_i \leq [\overline{x}_j])$$
$$V_U^*(j) \triangleq v(CPR_{\overline{\lambda}} | \overline{x}_j - \sum_{i \in N} \overline{a}_{ji} x_i \geq [\overline{x}_j]+1) .$$

Clearly
$$V_D^*(j) \leq v(CP | x_j \leq [\overline{x}_j])$$
$$V_U^*(j) \leq v(CP | x_j \geq [\overline{x}_j]+1)$$

for all $j \in I_f$, that is, these conditional bounds really do underestimate the optimal value of the candidate problems that would result if (CP) were separated using x_j as the separation variable.

Unfortunately the computation of $V_D^*(j)$ and $V_U^*(j)$ may be onerous if $\overline{a}_{ji} \neq 0$ for some variables $i \in N \cap I$. The computation then requires solving a knapsack-type problem with some integer variables. Hence it is natural to think of estimating $V_D^*(j)$ and $V_U^*(j)$ from below by simply dropping all integrality conditions:

(24)
$$V_D^{*o}(j) \triangleq v(\overline{CPR}_{\overline{\lambda}} | \overline{x}_j - \sum_{i \in N} \overline{a}_{ji} x_i \leq [\overline{x}_j])$$
$$V_U^{*o}(j) \triangleq v(\overline{CPR}_{\overline{\lambda}} | \overline{x}_j - \sum_{i \in N} \overline{a}_{ji} x_i \geq [\overline{x}_j]+1) .$$

The computation of each of these conditional bounds merely requires minimizing a linear function with all non-negative coefficients [$(c-\overline{\lambda}A) \geqq 0$ by duality] subject to a single linear constraint and $x \geqq 0$. This is sometimes referred to as a "continuous knapsack" type problem and it is easy to write down an explicit solution:

(25)

$$V_D^{*o}(j) = v(\overline{CP})+(\bar{x}_j - [\bar{x}_j]) \underset{\substack{i\in N: \\ \bar{a}_{ji}>0}}{\text{Minimum}} \left\{ \frac{(c-\bar{\lambda}A)_i}{\bar{a}_{ji}} \right\}$$

$$V_U^{*o}(j) = v(\overline{CP})+([\bar{x}_j]+1-\bar{x}_j) \underset{\substack{i\in N: \\ \bar{a}_{ji}<0}}{\text{Minimum}} \left\{ \frac{(c-\bar{\lambda}A)_i}{(-\bar{a}_{ji})} \right\}$$

(we have used the fact that $\bar{\lambda}b = v(\overline{CP})$ by LP duality). These conditional bounds are identical to those associated with the simplest penalties mentioned earlier (cf. (5). in [4]).

The strengthened penalties of Tomlin can also be recovered from this viewpoint by retaining the condition that x_j must not be in the open interval (0,1) for $j\in N\cap I$. Then we obtain

(26)

$$V_D^{*1}(j) \triangleq v(\overline{CPR}_{\bar{\lambda}} \,|\, \bar{x}_j - \sum_{i\in N} \bar{a}_{ji}x_i \le [\bar{x}_j] \text{ and } x_i \notin (0,1) \text{ for all } i\in N\cap I)$$

$$V_U^{*1}(j) \triangleq v(\overline{CPR}_{\bar{\lambda}} \,|\, \bar{x}_j - \sum_{i\in N} \bar{a}_{ji}x_i \ge [\bar{x}_j]+1 \text{ and } x_i \notin (0,1) \text{ for all } i\in N\cap I).$$

It is not difficult to see that at most one variable need be at a positive level in an optimal solution of the modified continuous knapsack problems defined in (26). This observation leads to the explicit formulae:

(27)

$$V_D^{*1}(j) = v(\overline{CP}) + \underset{\substack{i\in N: \\ \bar{a}_{ji}>0}}{\text{Min}} \left\{ \begin{array}{ll} (c-\bar{\lambda}A)_i(\bar{x}_j-[\bar{x}_j])/\bar{a}_{ji} & \text{if } i\notin I \\ (c-\bar{\lambda}A)_i \text{Max} \left\{ \dfrac{(\bar{x}_j-[\bar{x}_j])}{\bar{a}_{ji}}, 1 \right\} & \text{if } i\in I \end{array} \right\}$$

$$V_U^{*1}(j) = v(\overline{CP}) + \underset{\substack{i\in N \\ \bar{a}_{ji}<0}}{\text{Min}} \left\{ \begin{array}{ll} (c-\bar{\lambda}A)_i([\bar{x}_j]+1-\bar{x}_j)/(-\bar{a}_{ji}) & \text{if } i\notin I \\ (c-\bar{\lambda}A)_i \text{Max} \left\{ \dfrac{[\bar{x}_j]+1-\bar{x}_j}{(-\bar{a}_{ji})}, 1 \right\} & \text{if } i\in I \end{array} \right\}$$

These formulae are identical with the strengthened penalties of Tomlin (cf. (10) and (11) of [19] or (3.5) and (3.6) of [20]). It is evident from the very definitions (23),(24) and (26) that

$$\begin{aligned} &v_D^{*o}(j) \le v_D^{*1}(j) \le v_D^{*}(j) \\ (28) \qquad & \qquad\qquad\qquad\qquad \text{for all } j\epsilon I_f \,. \\ &v_U^{*o}(j) \le v_U^{*1}(j) \le v_U^{*}(j) \end{aligned}$$

This completes the recovery of known formulae for Driebeek and Tomlin penalties for $j\epsilon I_f$.
Exactly the same type of analysis holds for penalties associated with basic variables in $I-I_f$. Such penalties are of interest as a means of obtaining tighter ranges on integer variables which happen to be naturally integer in the optimal solution of $(\overline{CP})$. For a nonbasic variable x_j in $I-I_f$ the quantity of interest is $v(CPR_{\overline{\lambda}}|x_j \ge 1)$; no alternative representation in terms of nonbasic variables is possible. Evidently

$$(29) \qquad v(CPR_{\overline{\lambda}}|x_j \ge 1)=v(\overline{CP})+(c-\overline{\lambda}A)_j \,.$$

Again this is a standard result.

Another technique for strengthening (24) makes use of the following elementary result.

Theorem 3.

Let (IP) be a minimizing integer linear program in which exactly one variable, say x_h, is declared to be integer-valued. Suppose that $\overline{x}_h$, the optimal level of x_h when (IP) is solved ignoring the integrality requirement, is fractional. Then the optimal value of (IP) is given by

$$v(IP)=\text{Min}\ \{v(\overline{IP}|x_h=[\overline{x}_h]),v(\overline{IP}|x_h=[\overline{x}_h]+1)\}.$$

Proof. We have assumed (without loss of generality) that (IP) is a minimizing problem. Then it is well-known that $v(\overline{IP}|x_h)$ is a convex function of x_h with minimal value at $\bar{x}_h$. The desired conclusion follows easily. The possibility that $(\overline{IP}|x_h=[\bar{x}_h])$ or $(\overline{IP}|x_h=[\bar{x}_h]+1)$ or both are infeasible is not excluded (recall that our convention is to define a minimum over an empty set as $+\infty$).

Q.E.D.

Let $i_D(j)$ be the minimizing non-basic i in the formula for $v_D^{*o}(j)$ given in (25). The index $i_U(j)$ is defined similarly. Then application of Theorem 3 in the obvious way permits the following improvement on (24) to be computed with only a little extra effort:

$$v_D^{*2}(j) \triangleq v(\overline{CPR_\lambda}|\bar{x}_j - \sum_{i\in N} \bar{a}_{ji}x_i \le [\bar{x}_j] \text{ and } x_{i_D(j)} \text{ integer})$$

(30)

$$v_U^{*2}(j) \triangleq v(\overline{CPR_\lambda}|\bar{x}_j - \sum_{i\in N} \bar{a}_{ji}x_i \ge [\bar{x}_j]+1 \text{ and } x_{i_U(j)} \text{ integer}).$$

Neither (30) nor (26) necessarily dominates the other. It is not difficult to see that the following relationship then holds for $j\in I_f$:

$$(\bar{x}_j-[\bar{x}_j])/\ \bar{a}_{ji_D(j)}\begin{Bmatrix}\le\\ \ge\end{Bmatrix}1 \Rightarrow v_D^{*1}(j)\begin{Bmatrix}\ge\\ \le\end{Bmatrix}v_D^{*2}(j)$$

(31)

$$([\bar{x}_j]+1-\bar{x}_j)/(-\bar{a}_{ji_U(j)})\begin{Bmatrix}\le\\ \ge\end{Bmatrix}1 \Rightarrow v_U^{*1}(j)\begin{Bmatrix}\ge\\ \le\end{Bmatrix}v_U^{*2}(j).$$

We are unable to supply a reference to the conditional bounds (30) in the published literature.

So far we have required $Bx \geq d$ to be vacuous, that is, all upper bounds and other special constraints are treated as general A-type constraints. Analogs of the previous penalty results as well as new penalty results emerge easily by allowing $Bx \geq d$ to be non-vacuous. This will now be illustrated for the three examples.

The requirement made throughout this subsection that $Ax \geq b$ be replaced by $Ax = b$ in (P) served purely as a means of simplifying notation and facilitating comparison with equivalent results in the previously published literature. This requirement is now dropped.

5.2 Penalties for Example 1

Example 1 differs from the previous development only in that $(\text{CPR}_{\bar\lambda})$ now has upper-bounded variables. Both $V_D(j)$ and $V_U(j)$ still equal $v(\overline{\text{CP}})$ for all $j \in I_f$ by Theorem 2. The remedy for these vanishing penalties is again to invoke the representation for x_j which is available from the final LP tableau of $(\overline{\text{CP}})$. This representation will be written as

$$x_j = \alpha_{jo} - \sum_{i \neq j} \alpha_{ji} x_i \qquad \text{for } j \in I_f \tag{32}$$

where, of course, many of the coefficients α_{ji} may be 0. The resulting strengthened conditional lower bounds on $v(\text{CP}|x_j \leq [\bar{x}_j])$ and $v(\text{CP}|x_j \geq [\bar{x}_j] + 1)$ for $j \in I_f$ are

$$\begin{aligned} V_D^*(j) &\overset{\Delta}{=} v(\text{CPR}_{\bar\lambda}|x_j = \alpha_{jo} - \sum_{i \neq j} \alpha_{ji} x_i \leq [\bar{x}_j]) \\ V_U^*(j) &\overset{\Delta}{=} v(\text{CPR}_{\bar\lambda}|x_j = \alpha_{jo} - \sum_{i \neq j} \alpha_{ji} x_i \geq [\bar{x}_j] + 1). \end{aligned} \tag{33}$$

We have used the notation V_D^* and V_U^* as in (23) because (33) is an exact counterpart of (23). Like (23), (33) is generally computationally onerous because each estimate requires solving a knapsack-type problem in integer variables. The fact that the knapsack problem now has upper-bounded variables is but a small advantage. The most easily computed lower approximation to (33) is obtained by dropping the integrality requirements as in (24):

$$\begin{aligned} V_D^{*o}(j) &\overset{\Delta}{=} v(\overline{\text{CPR}_{\bar\lambda}}|x_j = \alpha_{jo} - \sum_{i \neq j} \alpha_{ji} x_i \leq [\bar{x}_j]) \\ V_U^{*o}(j) &\overset{\Delta}{=} v(\overline{\text{CPR}_{\bar\lambda}}|x_j = \alpha_{jo} - \sum_{i \neq j} \alpha_{ji} x_i \geq [\bar{x}_j] + 1). \end{aligned} \tag{34}$$

The notation v_D^{*o} and v_U^{*o} has again been carried over. The differences

$$(35) \qquad v_U^{*o}(j) - v(\overline{CP}) \text{ and } v_D^{*o}(j) - v(\overline{CP})$$

are Driebeek-like up and down penalties for Example 1. The computation of (34) requires only slightly more effort than the computation of (24). A "continuous knapsack" problem with upper-bounded variables must now be solved. Explicit formulae for v_D^{*o} and v_U^{*o} are slightly more cumbersome than expression (25), but are easily programmed for a digital computer.

To strengthen (34) one may formally write the counterpart of (26), but unfortunately explicit calculation may be nearly as costly as that of (33) itself. This is because the upper bounds generally invalidate the key property of (26) that at most one variable need be at a positive level in an optimal solution of each associated optimization problem. Thus the strengthened penalties of Tomlin do not generalize usefully to $Bx \geq d$ when it includes upper bounds on variables.

But the other technique based on Theorem 3 for strengthening $v_D^{*o}(j)$ and $v_U^{*o}(j)$ does generalize nicely. Let $i_D(j)$ and $i_U(j)$ be respectively the fractional-valued variables in the solutions of the optimizations corresponding to $v_D^{*o}(j)$ and $v_U^{*o}(j)$. It is easy to see that at most one variable need be fractional in each of these solutions; if none is, then that penalty cannot be strengthened by the present device. The strengthened conditional bounds analagous to (30) are:

$$V_D^{*2}(j) \overset{\Delta}{=} v(\overline{CPR_{\bar{\lambda}}} \mid x_j = \alpha_{jo} - \sum_{i \neq j} \alpha_{ji}\, x_i \leq [\bar{x}_j] \text{ and } x_{i_D(j)} \text{ integer})$$

(36)

$$V_U^{*2}(j) \overset{\Delta}{=} v(\overline{CPR_{\bar{\lambda}}} \mid x_j = \alpha_{jo} - \sum_{i \neq j} \alpha_{ji}\, x_i \geq [\bar{x}_j]+1 \text{ and } x_{i_U(j)} \text{ integer}).$$

The required optimizations are inexpensive to carry out on a digital computer. Clearly,

$$V_D^{*o}(j) \leq V_D^{*2}(j) \leq V_D^{*}(j) \leq v(CP \mid x_j \leq [\bar{x}_j])$$

(37)

$$V_U^{*o}(j) \leq V_U^{*2}(j) \leq V_U^{*}(j) \leq v(CP \mid x_j \geq [\bar{x}_j] + 1).$$

Exactly the same types of penalties can be constructed for $j \in I - I_f$ when the objective function coefficient of x_j vanishes in $(CPR_{\bar{\lambda}})$.

5.3 Penalties for Example 2

The development of penalties for Example 2 closely parallels that for Example 1. For $j \in I_f$ both up and down penalties again vanish by Theorem 2, and it is necessary to use the final LP tableau representation of the form (32).[1] The resulting conditional bounds $V_D^*(j)$ and $V_U^*(j)$ defined in (33) may still be too computationally onerous to use in general, but the multiple choice constraints do tend to make the computation easier by comparison with Example 1. One can imagine nontrivial situations where $V_D^*(j)$ and $V_U^*(j)$ can be computed relatively economically by a simple enumerative procedure. But in general one may have to fall back on the Driebeek-like penalties defined by (34). The required computations are no longer simple continuous knapsack problems with upper-bounded variables, but they can still be carried out efficiently

[1]Numbered displays from the discussion of Example 1 will be used here with the understanding that (CP), etc., have the structure of Example 2 rather than Example 1.

by specialized techniques (e.g., by parametric optimization applied to the dual of $(\overline{CPR_\lambda})$ with respect to the added constraint). Strengthening these penalties along the lines suggested by Tomlin as in (26) appears to be no easier in general than (33) itself. But again, as with Example 1, the strengthening (36) based on Theorem 3 is attractive. The indices $i_D(j)$ and $i_U(j)$ may be selected to be any of the fractional-valued variables in the solutions of the optimizations corresponding to $V_D^{*o}(j)$ and $V_U^{*o}(j)$. The implementation of (36) on a digital computer is only slightly more expensive than that of (34). Naturally, (37) continues to hold. The reader should have no difficulty seeing what to do if penalties are desired for variables in $I-I_f$ for reasons explained in Section 3.5.

The special nature of the multiple choice constraints (1) makes it possible to define "cumulative" conditional upper bounds as follows:

$$(38)\quad \begin{aligned} V_U^{*o}(j;J_k) &\overset{\Delta}{=} \text{Max}\left\{V_U^{*o}(j),\ V_D^{*o}(i) \text{ for } i \in \left\{J_k-j\right\}\right\} \\ V_U^{*2}(j;J_k) &\overset{\Delta}{=} \text{Max}\left\{V_U^{*2}(j),\ V_D^{*2}(i) \text{ for } i \in \left\{J_k-j\right\}\right\} \end{aligned}$$

where it is understood that j is in J_k in these definitions. That this provides true lower bounds on $v(CP|x_j = 1)$ follows from the fact that $x_j = 1$ implies $x_i = 0$ for all $i \neq j$ in the same multiple choice set.

5.4 Penalties for Example 3

For Example 3 we must partition I_f into two parts: $I_f \cap \{1,\ldots,K\}$ and $I_f - (I_f \cap \{1,\ldots,K\})$. For the first part, as described in detail in Section 3.4, the up and down penalties associated with $V_D(j)$ and $V_U(j)$ are highly unlikely to vanish. In fact, they are likely to be quite large. Thus it is unnecessary to invoke an alternative representation for such j from the final LP tableau. For $j \in I_f - (I_f \cap \{1,..,K\})$ however, it is easy to see that the naive penalties vanish and thus that the alternative representation may be useful. The detailed discussion would be so close to that for Example 2 that it will not be given here.

5.5 Penalties and Lagrangean cuts

We have seen that the Lagrangean relaxation viewpoint is consistent with standard penalty results, and that various improvements thereon for special problem structures such as Examples 1-3 then follow easily. What to do with these penalties has not been discussed. For this the reader should refer to Sections 3.4 and 3.5: the uses described there include all of the standard uses for penalties to be found in the published literature.

An additional use for penalties is in connection with Lagrangean cuts, via the options described in Section 4.5. Recall for Examples 1 and 2 that all cuts of type (11) have zero depth at $\bar{x}$ no matter which cut variables are selected. Both the simple penalties based on (34) and the strengthened penalties based on (36) can be used to generate cuts with positive depth at $\bar{x}$ so long as at least one of these penalties is non-zero. This may be done as follows. In order to keep the notation below consistent with that of Section 4, it will be assumed without loss of generality that (CP) equals (P) when applying the results of Sections 5.2-5.4.

Consider first the simple conditional bounds (34). Select any $j \in I_f$ such that at least one of the penalties (35) is strictly positive, and take this j to be the one cut variable. Clearly,

$$(39) \qquad v(\overline{PR}_\lambda | x_j) = \alpha_{jo} - \sum_{i \neq j} \alpha_{ji} x_i) \leq v(P | x_j)$$

holds for all fixed x_j, and in particular for all values of

x_j which are part of a feasible solution to (P). The left-hand side of (39) is convex as a function of x_j, and thus the unique linear function passing through it at the points $[\bar{x}_j]$ and $[\bar{x}_j] + 1$ must underestimate it for all integer values of x_j:

$$(40) \quad v_D^{*o}(j) + (v_U^{*o}(j) - v_D^{*o}(j))\ (x_j - [\bar{x}_j]) \leq$$

$$v(\overline{PR}_\lambda | x_j = \alpha_{jo} - \sum_{i \neq j} \alpha_{ji}\ x_i) \text{ for all integer } x_j .$$

Together (39) and (40) imply that

$$(41) \quad v_D^{*o}(j) + (v_U^{*o}(j) - v_D^{*o}(j))\ (x_j - [\bar{x}_j]) \leq cx$$

is a legitimate cut. This cut can be thought of as a relative of (11) which takes advantage of the two remarks in the closing paragraph of Section 4.5. Notice that if there are several $j \in I_f$ for which $v_D^{*o}(j)$ and $v_U^{*o}(j)$ are computed, then it is an easy matter to select j so as to yield the cut of type (41) which is deepest at $\bar{x}$.

Now consider the strengthene[illegible]onditional bounds (36). An analog of inequality (39) holds, but the analog of (40) does not because of the added integrality requirement in (36). It appears necessary to require that $j \in I_f$ be a 0-1 variable if a cut is to be based on (36) with j as the single cut variable. Then

$$(42) \quad v_D^{*2}(j) + (v_U^{*2}(j) - v_D^{*2}(j))\ x_j \leq cx$$

is a legitimate cut related to the original cut (11) by the last remark of Section 4.5. By (37), (42) is clearly a superior cut to (41). It is a simple matter to select j so as to yield the cut which is deepest at $\bar{x}$ among those of the form

For Example 2 one should of course use in (41) and (42) the cumulative penalties defined in (38) in place of $V_U^{*o}(j)$ or $V_U^{*2}(j)$ if the necessary quantities are at hand. One may further improve cuts (41) and (42) when j is a multiple choice variable by using one of the obvious cuts

(43) $$\sum_{j \epsilon J_k} V_U^{*o}(j;J_k)\ x_j \leq cx, \quad k=1,\ldots,K$$

or the still stronger cuts

(44) $$\sum_{j \epsilon J_k} V_U^{*2}(j;J_k)\ x_j \leq cx, \quad k=1,\ldots K.$$

Each of these cuts takes all of J_k as the set of cut variables, and is related to (11) by the last remark of Section 4.5. It is easy to verify that cuts of the form (43) [resp. (44)] are at least as strong as those of the form (41) [resp. (42)] for all x feasible in (p).

A cut very much in the spirit of (43) and (44) was proposed by Healy [15]. To be precise, for the k th cut he used $V_U^{*o}(j)$ as the coefficient of x_j, where $V_U^{*o}(j)$ is computed with $Bx \geq d$ taken to consist of only the k th multiple choice constraint (no upper bounds or other multiple choice constraints are included). This cut is legitimate for the same reason that (43) is, and is dominated by (43) because the coefficients of (43) are at least as large.

We note in closing that penalty-based cuts with more than one cut variable can often be obtained for Examples 1 and 2 and other structures by: (i) adding to $(PR_{\overline{\lambda}})$ relations of the form (32) for any subset of j's in I_f so long as no variable appears with a nonzero coefficient in more than one of these relations, and then (ii) exploiting separability. See the initial discussion of Section 4.5 for a development in much this same spirit. Details are left to the reader.

6. RELATIONS TO SURROGATE CONSTRAINTS

The surrogate constraint is a device which was originally developed in the context of pure 0-1 integer programs. There is nothing inherent in the nature of this device which limits its use to this context, as will become amply clear, but it will be more convenient to conduct our brief review of surrogate constraint ideas assuming until further notice that (P) is a purely 0-1 program with $Bx \geq d$ consisting solely of the unit upper bounds.

Glover [12] proposed that <u>surrogate</u> <u>constraints</u> of the form $\lambda(b-Ax) \leq 0$ with fixed $\lambda \geq 0$ be added to a 0-1 integer program in conjunction with its solution by an enumerative algorithm. Any $\lambda \geq 0$ leads to a surrogate constraint which is legitimate in the sense that it must be satisfied by all x feasible in (P), for it is merely a nonnegative linear combination of the given constraints. Such constraints were conceived primarily for use with enumerative algorithms that do not use an LP subroutine, for such algorithms have very limited ability to consider the conjoint implications of several constraints. Glover's criterion for the choice of λ was that it should maximize

$$\underset{x=0,1}{\text{Minimum}} \; cx \quad \text{subject to} \quad \lambda(b-Ax) \leq 0$$

over all $\lambda \geq 0$. Unfortunately, this criterion is prohibitively expensive to implement in general. By dropping the integrality requirements while keeping $0 \leq x \leq 1$, however, Balas [1] showed that the modified criterion permitted a characterization of the optimal λ as the dual solution $\bar{\lambda}$ corresponding to the constraints $Ax \geq b$ in the ordinary linear program $(\bar{P})$. This leads to a surrogate constraint that is identical to (13) with $\lambda=\bar{\lambda}$, which we remarked earlier is identical to Balas' "filter" constraint.

The present author took a different route in [8] by adding the constraint $cx < z^*$ to $\lambda(b-Ax) \leq 0$ [recall from Sec. 3.1 that z^* is the objective function value associated with the best known feasible solution to (P)] and rationalizing the following slightly different criterion for the choice of λ : maximize

$$\underset{x=0,1}{\text{Minimum}} \ \lambda(b-Ax)+(cx-z^*)$$

over $\lambda \geq 0$. This criterion also leads to $\bar{\lambda}$ as the optimal choice for λ . The resulting surrogate constraint advocated in [8] is

$$(45) \qquad cx+\bar{\lambda}(b-Ax) < z^* .$$

Two uses of (45) were proposed: as a possible means of fathoming via the easy test

$$(46) \qquad \left[\underset{x=0,1}{\text{Minimum}} \ cx+\bar{\lambda}(b-Ax)\right] \overset{?}{\geq} z^*$$

and as a possible means of range restriction via the following easy tests applied to the jth variable:

$$(47a) \qquad \left[\underset{x=0,1}{\text{Minimum}} \ \underset{\text{s.t. } x_j=0}{cx+\bar{\lambda}(b-Ax)}\right] \overset{?}{\geq} z^*$$

$$(47b) \qquad \left[\underset{x=0,1}{\text{Minimum}} \ \underset{\text{s.t. } x_j=1}{cx+\bar{\lambda}(b-Ax)}\right] \overset{?}{\geq} z^* .$$

If (46) holds then (P) is fathomed. If (47a) [resp. (47b)] holds then x_j must be 1 [resp. 0] in any feasible solution of (P) which is superior in value to the current incumbent. It is important to add that not only would these tests be performed as soon as (45) is created, but also subsequently for candidate problems derived from (P) by separation -- in which case the accumulated separation constraints would be appended to the optimizations in (46) and (47). Note that this does not spoil the essential triviality of (46) and (47) from a computational standpoint.

It is easy to interpret (46) and (47) from the viewpoint of Lagrangean relaxation with $Bx \geq d$ consisting of the upper bound constraints $x_j \leq 1$. Test (46) can be rewritten as

$$(48) \qquad v(PR_{\bar{\lambda}}) \overset{?}{\geq} z^* ,$$

which is precisely the elementary fathoming criterion associated with $(PR_{\bar{\lambda}})$. Similarly, (47) can be rewritten as

$$(49a) \qquad v(PR_{\bar{\lambda}} | x_j = 0) \overset{?}{\geq} z^*$$

$$(49b) \qquad v(PR_{\bar{\lambda}} | x_j = 1) \overset{?}{\geq} z^* .$$

This is precisely the range restriction criterion described in Sec. 3.5. When (46) and (47) are performed on subsequent candidate problems derived from (P) by separation, (48) and (49) are still standard Lagrangean relaxation tests except that an "old" λ is used in (CPR_{λ}). This is the standard "pre-LP" strategy mentioned in Sec. 3.7.

Thus the surrogate constraint (45) and the tests based on it are seen to be completely subsumed by the simplest Lagrangean relaxation techniques. Generalizations of (45) - (47) when (P) is not a pure 0-1 program or when $Bx \geq d$ includes more than simple upper bounds can be obtained without difficulty. Some such generalizations were developed a few years ago by this writer in unpublished lecture notes and by Glover [13] using the surrogate constraint viewpoint, but in every case the same results are easy special cases of more general and powerful results based on Lagrangean relaxation. It therefore appears that the surrogate constraint viewpoint can be discarded in favor of the Lagrangean relaxation viewpoint.

7. CONCLUSION

We hope that the length of this development has not obscured the basic ideas that we have tried to convey. Foremost among these is the usefulness of Lagrangean relaxation as a unified viewpoint from which one may develop auxiliary devices to improve the performance of existing branch-and-bound algorithms for integer programming (Sec. 3), develop new cutting-planes (Sec. 4), develop new penalties (Sec. 5), and supplant the narrower notions of surrogate constraints (Sec. 6) and relaxation by simply discarding troublesome constraints. All of these things usually can be done in a way that exploits the special structure of the particular problem class at hand. The trick is to judiciously choose the subset of constraints to play the role of $Bx \geq d$ in our statement of the general problem (P). The three examples of Sec. 1.1 were carried throughout the paper to indicate how flexible the application of Lagrangean relaxation can be, and how dependent the results are on the particular choice of $Bx \geq d$. The Integrality Property first encountered in Theorem 1 appears to yield a useful dichotomy of the universe of possible choices for $Bx \geq d$; choices satisfying this property seem to have less inherent potential than choices not satisfying it, and often require extra measures to achieve attractiveness (this is the main subject of Sec. 5).

There is much work yet to be done to develop the full potential usefulness of Lagrangean relaxation for various special problem classes. Several basic strategies require further exploration, as discussed in Sec. 3.7. The recent work in this vein cited earlier [5] [6] [7] [16] [17] is an auspicious beginning, but is still just a beginning.

To this list can be added the current work of the author [10] on warehouse location problems with lower as well as upper capacities on warehouse throughput, and with arbitrary additional constraints. Such problems require but a slight generalization of Example 3 in this paper. The algorithm of [8] (see also Sec. 3.1.5 of [11]) has been modified to use both cuts and penalties based on Lagrangean relaxation. Preliminary computational experience with several practical problems suggests that the first Lagrangean cut is much more effective than Gomory's 1960 mixed-integer cut; and that the Lagrangean penalties are typically at least an order of magnitude higher than Tomlin's strengthened penalties, yet less expensive to compute. The net result is that overall computing times are greatly reduced. We feel that this experience alone justifies further serious investigation for other significant classes of problems.

REFERENCES

1. Balas, E., "Discrete Programming by the Filter Method", Operations Research, Vol. 15, No. 5 (September-October 1967), pp. 915-957.

2. Brooks, R. and Geoffrion, A., "Finding Everett's Lagrange Multipliers by Linear Programming", Operations Research, 14, 6 (November-December 1966), pp. 1149-1153.

3. Dakin, R.J., "A Tree Search Algorithm for Mixed Integer Programming Problems", Computer Journal, Vol. 8, No. 3 (1965), pp. 250-255.

4. Driebeek, N.J., "An Algorithm for the Solution of Mixed Integer Programming Problems", Management Science, Vol. 12, No. 7 (March 1966), pp. 576-587.

5. Fisher, M.L., "Optimal Solution of Scheduling Problems Using Lagrange Multipliers: Part I", Report 7210, July 1972, Center for Mathematical Studies in Business and Economics, University of Chicago. To appear in Operations Research.

6. ______ and Schrage, L., "Using Lagrange Multipliers to Schedule Elective Hospital Admissions", Working Paper, July 1972, University of Chicago.

7. _____ and Shapiro, J.F., "Constructive Duality in Integer Programming", Report 7224, May 1972, Center for Mathematical Studies in Business and Economics, University of Chicago.

8. Geoffrion, A.M., "An Improved Implicit Enumeration Approach for Integer Programming", Operations Research, Vol. 17, No. 3 (May-June 1969), pp. 437-454.

9. _____, "Duality in Nonlinear Programming", SIAM Review, 13, 1 (January 1971), pp. 1-37.

10. _____, "The Capacitated Plant Location Problem with Additional Constraints", paper presented to the Joint National Meeting of AIIE, ORSA, and TIMS, Atlantic City, November 8-10, 1972.

11. _____ and Marsten, R.E., "Integer Programming Algorithms: A Framework and State-of-the-Art Survey", Management Science, 18, 9 (May 1972), pp. 465-491.

12. Glover, F., "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem", Operations Research, Vol. 13, No. 6 (November-December 1965), pp. 879-919.

13. _____, "Surrogate Constraints", Operations Research, 16, 4 (July-August 1968), pp. 741-749.

14. Greenberg, H.J. and Robbins, T.C., "Finding Everett's Lagrange Multipliers by Generalized Linear Programming, Parts I, II, and III", Technical Report CP-70008, revised June 1972, Computer Science/Operations Research Center, Southern Methodist University.

15. Healy, W.C., Jr., "Multiple Choice Programming", Operations Research, 12, 1 (January-February 1964), pp. 122-138.

16. Held, M. and Karp, R.M., "The Traveling Salesman Problem and Minimum Spanning Trees", Operations Research, Vol. 18, No. 6 (November-December 1970), pp. 1138-1162.

17. ______ and _____, "The Traveling Salesman Problem and Minimum Spanning Trees: Part II", Mathematical Programming, 1, 1 (October 1971), pp. 6-25 .

18. Shapiro, J.F., "Generalized Lagrange Multipliers in Integer Programming", Operations Research, 19, 1 (January-February 1971), pp. 68-76.

19. Tomlin, J.A., "An Improved Branch and Bound Method for Integer Programming", Operations Research, Vol. 19, No. 4 (July-August 1971), pp. 1070-1075 .

20. _____, "Branch and Bound Methods for Integer and Non-Convex Programming", in J. Abadie (ed.), Integer and Nonlinear Programming, North-Holland, Amsterdam, 1970.

21. Veinott, A.F. and Dantzig, G.B., "Integral Extreme Points", SIAM Review, 10, 3 (July 1968), pp. 371-372.